Your search matched **19809** of **1006282** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enter new one in the text box.

lexical <or> search <or> tree and signatures Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Band search: an efficient alternative to guided depth-first search**
Chu, L.-C.; Wah, B.W.;
Tools with Artificial Intelligence, 1992. TAI '92, Proceedings., Fourth Internati Conference on , 10-13 Nov. 1992
Pages:154 - 161

[Abstract] [PDF Full-Text (644 KB)] **IEEE CNF**

---

2 **Lattice-based search strategies for large vocabulary speech recogni**
Richardson, F.; Ostendorf, M.; Rohlicek, J.R.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 Internatio Conference on , Volume: 1 , 9-12 May 1995
Pages:576 - 579 vol.1

[Abstract] [PDF Full-Text (380 KB)] **IEEE CNF**

---

3 **An efficient search algorithm for BLOCK motion estimation**
Jae-Yong Kim; Sung-Bong Yang;
Signal Processing Systems, 1999. SiPS 99. 1999 IEEE Workshop on , 20-22 O 1999
Pages:100 - 109

[Abstract] [PDF Full-Text (692 KB)] **IEEE CNF**

---

4 **Optimal and heuristic search for a hidden object in one dimension**
Cox, Z.A., Jr.; Xiaorong Sun; Yuping Qiu;
Systems, Man, and Cybernetics, 1994. 'Humans, Information and Technology 1994 IEEE International Conference on , Volume: 2 , 2-5 Oct. 1994
Pages:1252 - 1256 vol.2

h eee e eee g e ch e ch e e c e e e c e

[Abstract] [PDF Full-Text (427 KB)] IEEE CNF

5 **A meta-search method reinforced by cluster descriptors**
Yipeng Shen; Dik Lun Lee;
Web Information Systems Engineering, 2001. Proceedings of the Second International Conference on , Volume: 1 , 3-6 Dec. 2001
Pages:125 - 132 vol.1

[Abstract] [PDF Full-Text (763 KB)] IEEE CNF

6 **On the complexity of search algorithms**
Kuo-Liang Chung; Wen-Chin Chen; Ferng-Ching Lin;
Computers, IEEE Transactions on , Volume: 41 , Issue: 9 , Sept. 1992
Pages:1172 - 1176

[Abstract] [PDF Full-Text (356 KB)] IEEE JNL

7 **The rendezvous and coordinated search problems**
Alpern, S.; Reyniers, D.J.;
Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on , Vc 1 , 14-16 Dec. 1994
Pages:513 - 517 vol.1

[Abstract] [PDF Full-Text (356 KB)] IEEE CNF

8 **A new multilevel codebook searching algorithm for vector quantizat**
Cao, H.Q.; Li, W.;
Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on , Volume: 1 , 28 April-3 May 1995
Pages:441 - 444 vol.1

[Abstract] [PDF Full-Text (376 KB)] IEEE CNF

9 **Efficient motion estimation using multiple log searching and adapti search windows**
Marlow, S.; Ng, J.; McArdle, C.;
Image Processing and Its Applications, 1997., Sixth International Conference on , Volume: 1 , 14-17 July 1997
Pages:214 - 218 vol.1

[Abstract] [PDF Full-Text (384 KB)] IEE CNF

10 **Tuning up the search engine**
Williams, J.; Starzl, R.;
IT Professional , Volume: 3 , Issue: 3 , May-June 2001
Pages:60 - 62

[Abstract] [PDF Full-Text (264 KB)] IEEE JNL

11 **A tree search strategy for large-vocabulary continuous speech recognition**
Gopalakrishnan, P.S.; Bahl, L.R.; Mercer, R.L.;
Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 Internatio

h eee e eee g e ch e ch e e c e e e c e

Conference on , Volume: 1 , 9-12 May 1995
Pages:572 - 575 vol.1

[Abstract] [PDF Full-Text (304 KB)] IEEE CNF

---

12 **An efficient search strategy for block motion estimation using ima features**
Yui-Lam Chan; Wan-Chi Siu;
Image Processing, IEEE Transactions on , Volume: 10 , Issue: 8 , Aug. 2001
Pages:1223 - 1238

[Abstract] [PDF Full-Text (392 KB)] IEEE JNL

---

13 **Cellular search algorithm for motion estimation**
Jeanson Hung; Wen-Sheng Su; Jun-Hua Wang;
Digital and Computational Video, 2001. Proceedings. Second International Workshop on , 8-9 Feb. 2001
Pages:173 - 179

[Abstract] [PDF Full-Text (276 KB)] IEEE CNF

---

14 **An agent-based search engine based on the Internet search servic the CORBA**
Yue-Shan Chang; Hsin-Chun Hsich; Shyan-Ming Yuan; Lo, W.;
Distributed Objects and Applications, 1999. Proceedings of the International Symposium on , 5-6 Sept. 1999
Pages:26 - 33

[Abstract] [PDF Full-Text (124 KB)] IEEE CNF

---

15 **Enhancing conventional search systems with multi-agent techniqu case study**
Denzinger, J.; Fuchs, D.;
Multi Agent Systems, 1998. Proceedings. International Conference on , 3-7 Ju 1998
Pages:419 - 420

[Abstract] [PDF Full-Text (24 KB)] IEEE CNF

---

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 2
25 26 27 28 29 30 31 32 33 34 Next

h eee e eee g e ch e ch e e c e e e c e

US-PAT-NO: **5319779**

DOCUMENT-IDENTIFIER: US 5319779 A
See image for Certificate of Correction

TITLE: System for **searching** information using combinatorial signature derived from bits sets of a base signature

---------- KWIC ---------

Abstract Text - ABTX (1):
This invention encodes information (such as the field values of a database record, or the **words** of a text document) so that the original information may be efficiently **searched** by a computer. An information object is encoded into a small "signature" or codeword using a method. A base or "leaf" signature S1 34 is computed by a known technique such as hashing. The logical intersection (AND) of each possible combination of pairs of bits of the base signature is computed, and the result is stored as one bit of a longer combinatorial signature CS1 42. The bit-wise logical union (bit-OR) of the combinatorial signatures of a group of records produces a second-level combinatorial signature CS2 52 representing particular field values present among those records. Higher-level combinatorial signatures CS3 60, CS4, etc. are computed similarly. These combinatorial signatures avoid a "saturation" problem which occurs when signatures are grouped together, and a "combinatorial error" problem which falsely indicates the existence of nonexistent records, thereby significantly improving the ability to reject data not relevant to a given

Google™ [lexical search tree and matching] [Google Search] Advanced Search Preferences Language Tools Search Tips

The "AND" operator is unnecessary -- we include all search terms by default. [details]

Web · Images · Groups · Directory · News ·

Searched the web for **lexical search tree and matching characters and ASCII and hash value and nodes**. Re

[PPT] **Lexical** Analysis and Scanning
File Format: Microsoft Powerpoint 97 - View as HTML
... Chain according to **hash** code. Serial **search** on one chain. ... Parser builds **tree** incrementally, using successive tokens as **tree nodes**. ... Chapter 2. **Lexical** Elements. ...
www.cs.nyu.edu/courses/spring04/G22.2130-001/lex.ppt - Similar pages

whatis - list of manual pages
... analyzer generator flexdoc (1) - fast **lexical** analyzer generator ... directory for transmission grep (1) - **search** a file ... checksums of the entire source **tree** sync (8 ...
www.cs.vu.nl/pub/minix/2.0.0/wwwman/whatis.html - 31k - Feb 22, 2004 - Cached - Similar pages

bc (1) - An arbitrary precision calculator language cc (1) - C ...
... **lexical** analyzer generator flexdoc (1) - fast **lexical** analyzer generator ... of a file grep (1) - **search** a file ... CRC checksums of the entire source **tree** add_route (8 ...
www.cs.vu.nl/pub/minix/2.0.0/manuals/WHATIS - 18k - Cached - Similar pages
[More results from www.cs.vu.nl]

PMSI - index by author
... Give me something $self-ish. **Lexical** aliasing sub params. perl -x to test while developing. ... Pig Latin. tie class for **Search**::Dict. ... **ASCII** Christmas **tree** generator. ...
grinder.perlmonk.org/pmsi/author.html - 101k - Cached - Similar pages

ActivePython 2.2 - Online Docs
... 1 Regular Expression Syntax; 4.2.2 **Matching** vs **Searching**; ... 1 Cmd Objects. 5.15 shlex -- Simple **lexical** analysis: ... Constants used with Python parse **trees**; 18.4 keyword ...
aspn.activestate.com/ASPN/docs/ ActivePython/2.2/python/lib/ - 68k - Cached - Similar pages

Programming Tools (Page 31 of 88) - WindowsPC.com
... Pattern String Engine is intended for **lexical** analysis of ... new high efficient data structure for fast **searching**. ... hierarchical information as a graphs **tree** on an ...
www.windowspc.com/prog31.htm - 32k - Cached - Similar pages

Perl Modules - Perl Doc at IceWalkers.com
... Lint Perl lint B::Showlex Show **lexical** variables used ... scalar subroutines **Search**::Dict, look **search** for key ... A perl module for querying XML **tree** structures with ...
www.icewalkers.com/Perl/5.8.0/lib.html - 61k - Cached - Similar pages

Protocols:
... for example, avoids broadcasts through a clever **lexical** routing scheme ... fact that this network is a **tree** is a ... protocol, this scheme limits your **search** radius to ...
www.limewire.com/developer/query_routing/ keyword%20routing.htm - 101k - Cached - Similar pages

[PDF] **Lexical** Discovery with an Enriched Semantic Network
File Format: PDF/Adobe Acrobat - View as HTML
... sequences Exploring the WordNet hyponym **tree** starting at ... by simply performing breadth rst **search** on the ... a hard coded path con straint **Lexical** FreeNet **Lexical** ...
www.lexfn.com/doc/lexfn.pdf - Similar pages

h g gec e ch h e e e e c e ch ee

[PS] Automized Generation of Typed Syntax **Trees** via XML Baltasar Tranc' ...
File Format: Adobe PostScript - View as Text
... given, and a rule can be marked as a **lexical** category, containing ... many other kinds of processing of the TDOM **tree**,such as **searching**, sorting, extracting ...
uebb.cs.tu-berlin.de/papers/published/xse2001.ps - Similar pages

Goooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

lexical search tree and matching | Google Search | Search within results

Dissatisfied with your search results? Help us improve.

Get the Google Toolbar: 

h g gec e ch h e e e e c e ch ee

	Type	Hits	Search Text	DBs
1	BRS	3709	BINARY NEAR2 TREE	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
2	BRS	37	(BINARY NEAR2 TREE) and ((invert or reverse) near2 tree)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
3	BRS	3	((BINARY NEAR2 TREE) and ((invert or reverse) near2 tree)) and pointers	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
4	BRS	4208	707/100	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
5	BRS	30	707/100 and (sort$2 near2 tree)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
6	BRS	2	5369577.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
7	BRS	1	5369577.pn. and ASCII	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
8	BRS	0	5319779.pn. and lexical	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
9	BRS	2	5319779.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
10	BRS	0	5319779.pn. and lexical	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
11	BRS	2955908	5369577.pn.l	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
12	BRS	2	5369577.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
13	BRS	1	5369577.pn. and hash	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
14	BRS	19	(lexical near2 search) same tree	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
15	BRS	7	((lexical near2 search) same tree) and characters	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
16	BRS	4	((lexical near2 search) same tree) and characters	USPAT; EPO; JPO; DERWENT; IBM_TDB
17	BRS	9	((lexical near2 search) same tree).ab.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
18	BRS	2	(((lexical near2 search) same tree).ab.) and ASCII	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
19	BRS	30	(lexical near2 search) and tree	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
20	BRS	1	((lexical near2 search) and tree) and ASCII	USPAT; EPO; JPO; DERWENT; IBM_TDB
21	BRS	1	5319779.pn. and hash	USPAT; EPO; JPO; DERWENT; IBM_TDB
22	BRS	1	5319779.pn. and (hash same signature)	USPAT; EPO; JPO; DERWENT; IBM_TDB
23	BRS	2	5319779.pn. and (search$ near$4 word$)	USPAT; EPO; JPO; DERWENT; IBM_TDB

	Type	Hits	Search Text	DBs
238	BRS	10	707/5$.ccls. and ((document near2 size) same range)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
239	BRS	6	(707/5$.ccls. and ((document near2 size) same range)) and weight	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
240	BRS	2	6353825.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
241	BRS	1	6353825.pn. and size	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
242	BRS	522	((document near2 size) same range)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
243	BRS	7	(((document near2 size) same range)) and (range near5 weight)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
244	BRS	2	5650941.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
245	BRS	14	707/5$.ccls. and ((document near5 size) same range)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
246	BRS	2	09/998682	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
247	BRS	2	5202986.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
248	BRS	2	5202986.pn. and (key same index)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
249	BRS	2	5202986.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
250	BRS	1	5202986.pn. and ((root near2 node) same index)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
251	BRS	1	5202986.pn. and (creat$ near4 branch)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
252	BRS	0	5202986.pn. and (creat$ near4 subtree)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
253	BRS	2	5202986.pn. and subtree	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
254	BRS	1	5202986.pn. and ((leaf near2 node) with key)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
255	BRS	2	5369577.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
256	BRS	2	5369577.pn. and stem	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
257	BRS	1	5202986.pn. and substring	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
258	BRS	2	5202986.pn. and string	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
259	BRS	2	5202986.pn. and character	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
260	BRS	1	5369577.pn. and substring	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
261	BRS	1	5369577.pn. and character	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
262	BRS	1	5369577.pn. and (character same node)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
263	BRS	1	5369577.pn. and (root same index)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
264	BRS	1	5369577.pn. and (location same root)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
265	BRS	0	5369577.pn. and (allocation same node)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
266	BRS	1	5369577.pn. and branch	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
267	BRS	1	5369577.pn. and (index same stem)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB

	Type	Hits	Search Text	DBs
268	BRS	1	5369577.pn. and (index same stem) and (index near4 value)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
269	BRS	0	5369577.pn. and branh	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
270	BRS	1	5369577.pn. and branch	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
271	BRS	0	5369577.pn. and (create near2 branch)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
272	BRS	1	5369577.pn. and subtree	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
273	BRS	1	5369577.pn. and empty	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
274	BRS	1	5369577.pn. and branch	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
275	BRS	1	10/002381	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
276	BRS	0	5369577.pn. and (insert$ near2 node)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
277	BRS	1	5369577.pn. and (creat$ near2 node)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB
278	BRS	1	5369577.pn. and (index same stem)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB